Light Control Circuit Design with Clock Division

Sl No	Name	ID No
1	**Kaushik Kalakonda**	**2020A3PS0468H**

Problem Statement: Design a state machine to control the tail lights of a 1965 Ford Thunder bird (Figure 1). The tail lights are composed of three lights on each side which operate for the turns as shown in figure 2. The state machine has two inputs **(LEFT, RIGHT)** and 6 outputs **(LC, LB, LA, RA, RB and RC)**. When **(RIGHT=0 and LEFT=0)** or when **(RIGHT=1 and LEFT=1)** no lights will turn ON. If **(RIGHT=0 and LEFT=1)** then lights LC, LB, and LA will be ON as shown in figure 2(a) indicating **LEFT** turn. If **(RIGHT=1 and LEFT=0)** then lights RA, RB, and RC will be ON as shown in figure 2(b) indicating **RIGHT** turn. In addition to LEFT and RIGHT there are two more inputs **Clk** and **Reset** for normal operation of FSM. When **Reset** is enabled all lights will be OFF**.** The flashing rate of LEDs is 2Hz (or 4Hz) (i.e. the time between two successive states is 0.5s (or 0.25s)).

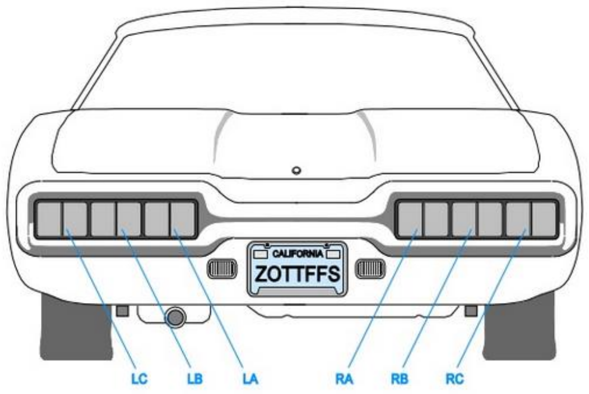


Figure 1

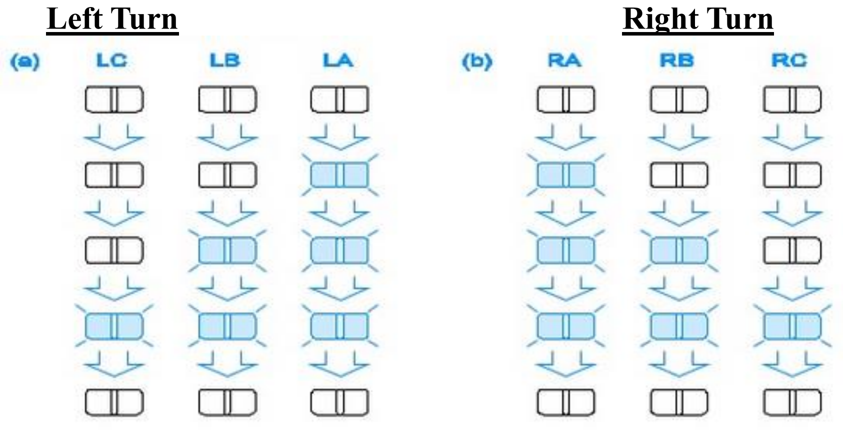


Figure 2(a) **Figure 2(b)**

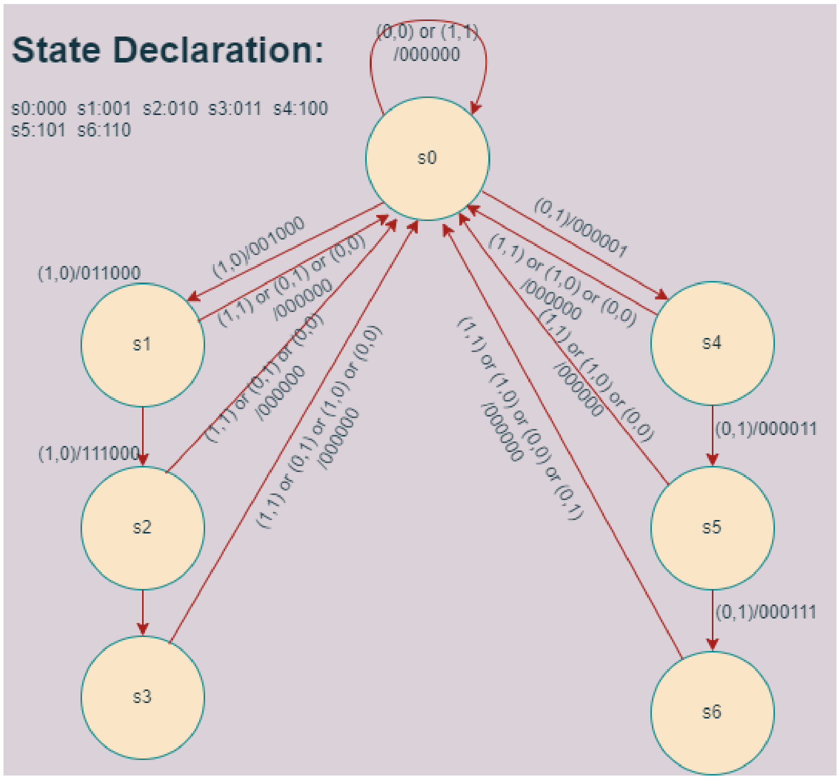
Figure 2

(Please refer to the file named "**Vivado_Design_Flow_All_Steps.pdf**" for a review of all the steps in the design flow)

1. **Question: Draw the FSM (can be an image) with proper description.**

   Answer:

   Mealy representation of FSM

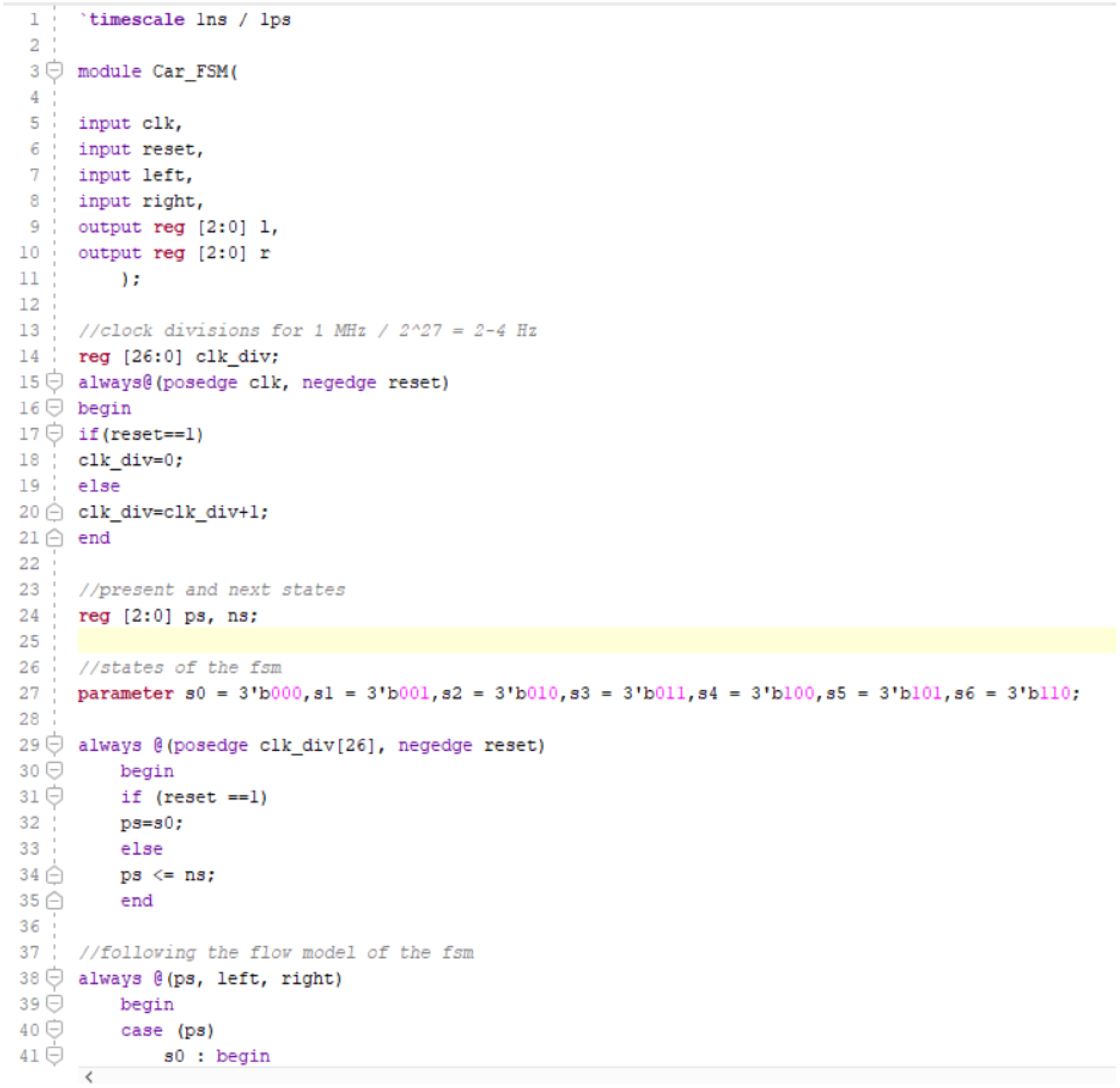


State Declaration:
s0:000 s1:001 s2:010 s3:011 s4:100
s5:101 s6:110
(0,0) or (1,1) /000000
s0
(1,0)/001000
(1,1) or (0,1) or (0,0) /000000
(1,0)/011000
s1
(1,1) or (0,1) or (0,0) /000000
(1,0)/111000
s2
(1,1) or (0,1) or (1,0) or (0,0) /000000
s3
(0,1)/000001
(1,1) or (1,0) or (0,0) /000000
s4
(0,1)/000011
(1,1) or (1,0) or (0,0) /000000
s5
(0,1)/000111
(1,1) or (1,0) or (0,0) or (0,1) /000000
s6

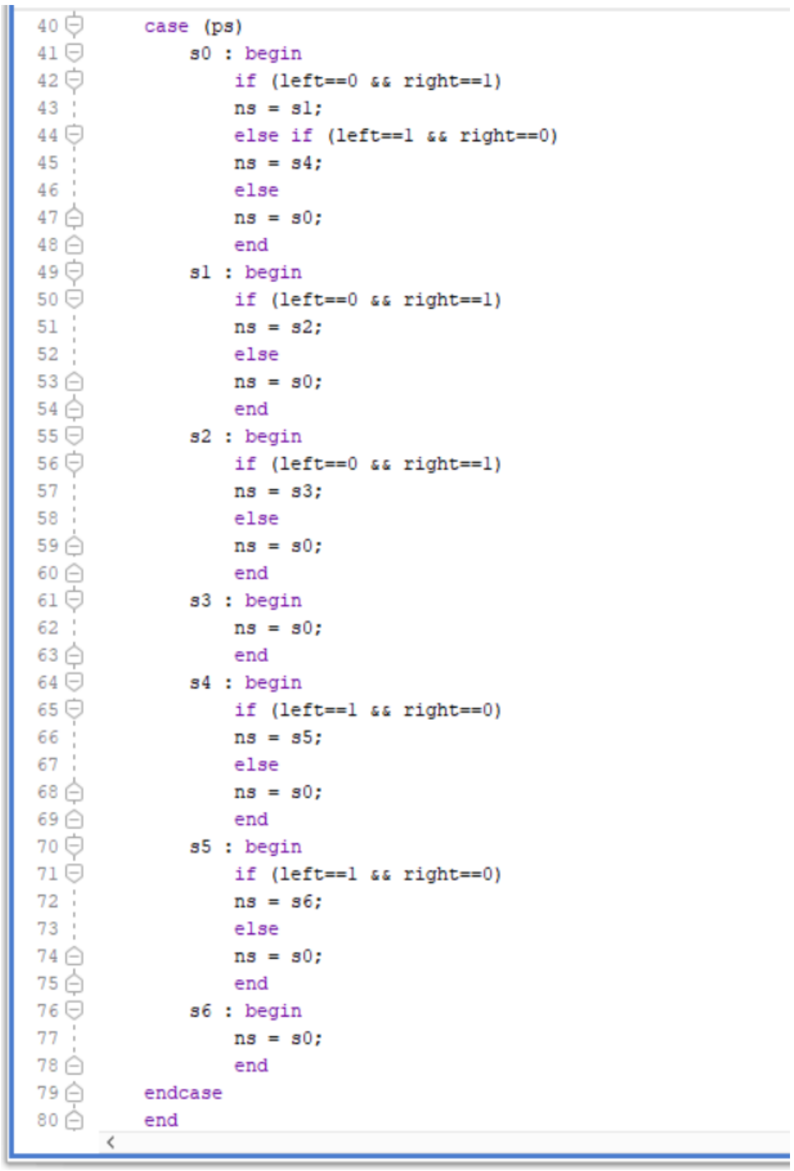
2. Create a Vivado Project and write Verilog code (**Car_FSM.v** with comments) for implementing the above FSM.

   Question: Paste the image of verilog code Car_FSM.v.

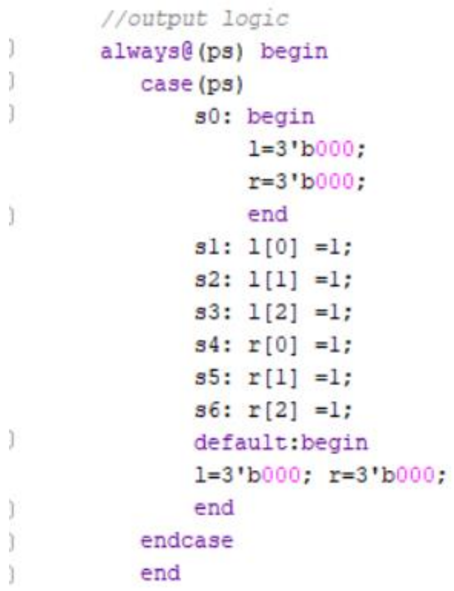
Answer:



```verilog
`timescale 1ns / 1ps

module Car_FSM(

input clk,
input reset,
input left,
input right,
output reg [2:0] l,
output reg [2:0] r
    );

//clock divisions for 1 MHz / 2^27 = 2-4 Hz
reg [26:0] clk_div;
always@(posedge clk, negedge reset)
begin
if(reset==1)
clk_div=0;
else
clk_div=clk_div+1;
end

//present and next states
reg [2:0] ps, ns;

//states of the fsm
parameter s0 = 3'b000,s1 = 3'b001,s2 = 3'b010,s3 = 3'b011,s4 = 3'b100,s5 = 3'b101,s6 = 3'b110;

always @(posedge clk_div[26], negedge reset)
    begin
    if (reset ==1)
    ps=s0;
    else
    ps <= ns;
    end

//following the flow model of the fsm
always @(ps, left, right)
    begin
    case (ps)
        s0 : begin
```



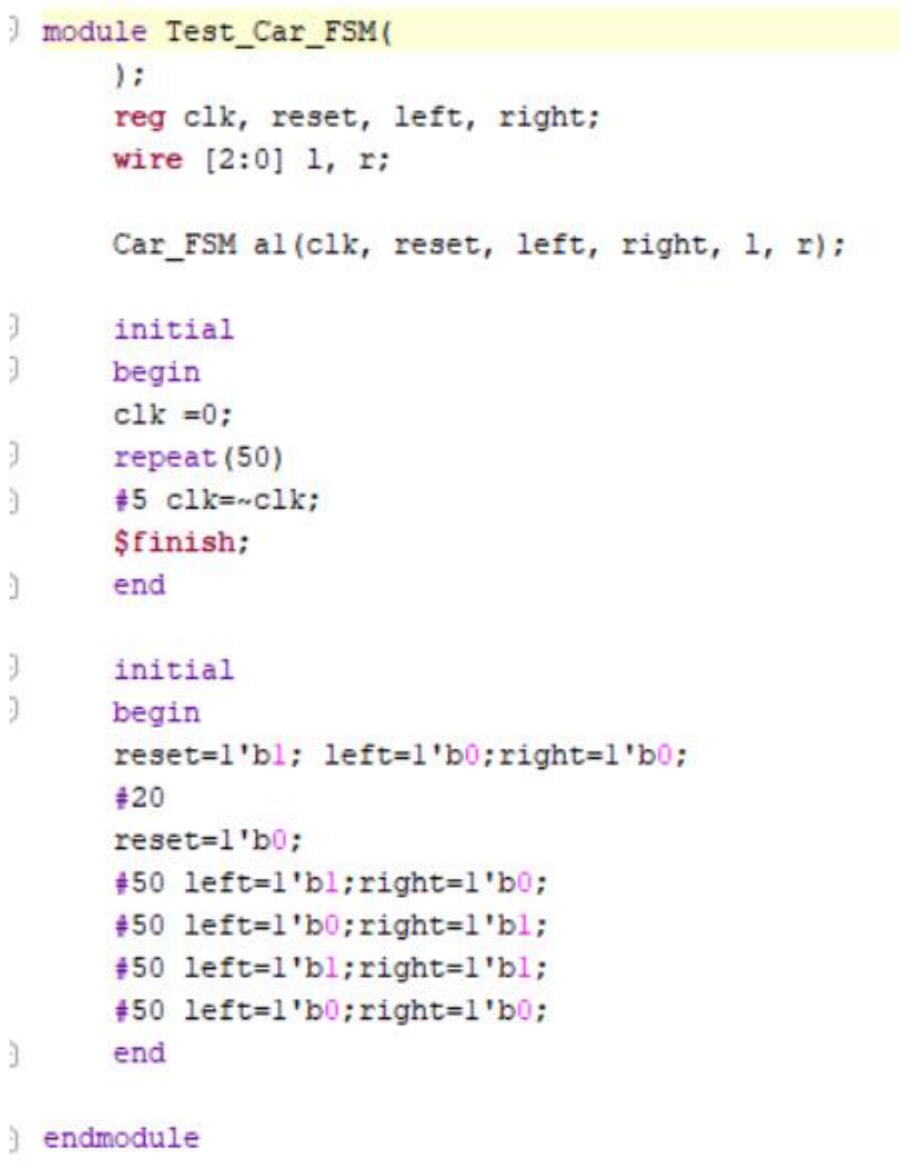
```
case (ps)
    s0 : begin
        if (left==0 && right==1)
        ns = s1;
        else if (left==1 && right==0)
        ns = s4;
        else
        ns = s0;
        end
    s1 : begin
        if (left==0 && right==1)
        ns = s2;
        else
        ns = s0;
        end
    s2 : begin
        if (left==0 && right==1)
        ns = s3;
        else
        ns = s0;
        end
    s3 : begin
        ns = s0;
        end
    s4 : begin
        if (left==1 && right==0)
        ns = s5;
        else
        ns = s0;
        end
    s5 : begin
        if (left==1 && right==0)
        ns = s6;
        else
        ns = s0;
        end
    s6 : begin
        ns = s0;
        end
endcase
end
```



```
//output logic
always@(ps) begin
   case(ps)
       s0: begin
           l=3'b000;
           r=3'b000;
           end
       s1: l[0] =1;
       s2: l[1] =1;
       s3: l[2] =1;
       s4: r[0] =1;
       s5: r[1] =1;
       s6: r[2] =1;
       default:begin
       l=3'b000; r=3'b000;
       end
   endcase
   end
```

3. Write the test bench **Test_Car_FSM.v** and simulate your design to check the functionality.

Question: Paste the image of test bench verilog code Test_Car_FSM.v.

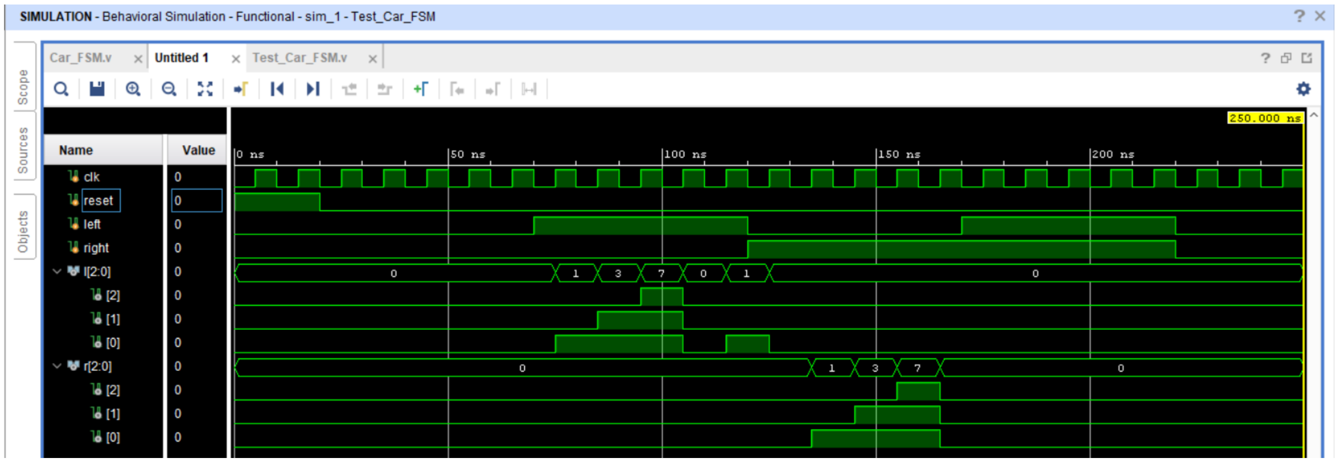


```
module Test_Car_FSM(
    );
    reg clk, reset, left, right;
    wire [2:0] l, r;

    Car_FSM a1(clk, reset, left, right, l, r);

    initial
    begin
    clk =0;
    repeat(50)
    #5 clk=~clk;
    $finish;
    end

    initial
    begin
    reset=1'b1; left=1'b0;right=1'b0;
    #20
    reset=1'b0;
    #50 left=1'b1;right=1'b0;
    #50 left=1'b0;right=1'b1;
    #50 left=1'b1;right=1'b1;
    #50 left=1'b0;right=1'b0;
    end

endmodule
```

Answer:

Question: Paste the image showing the simulated waveforms for FSM (Behavioral Simulation). Clearly show the LEFT turn and RIGHT turn Cases.

Answer:



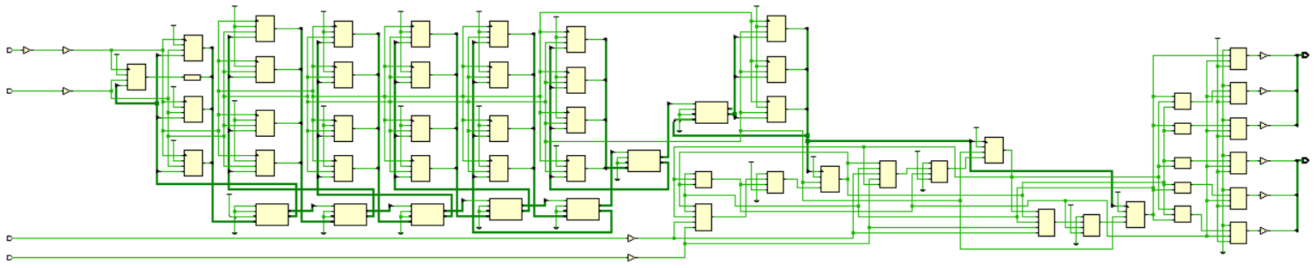
4. Add clock division code to **Car_FSM.v** such that the actual input **Clk (Y9 pin with frequency of 100MHz)** is converted to Clock of frequency 2Hz (or 4Hz). This 2Hz (or 4Hz) signal is used as clock for running the FSM. Connect the new (2Hz or 4Hz)) clock as output of **Car_FSM.v** for reference.

5. Plan your I/O mapping (using **I/O planning** option) such that actual input **Clk** is connected to internal clock pin **Y9**, **Reset** is connected to push button switch, other inputs (**LEFT** and **RIGHT**) are connected to DIP switches and outputs are connected to LEDs. In the ZedBoard, the pin numbers indicating the DIP switches, LEDs and internal clock are listed in table uploaded in CMS. Save the mapping information as **Car_FSM.xdc**.

6. Synthesize (**Run Synthesis**).

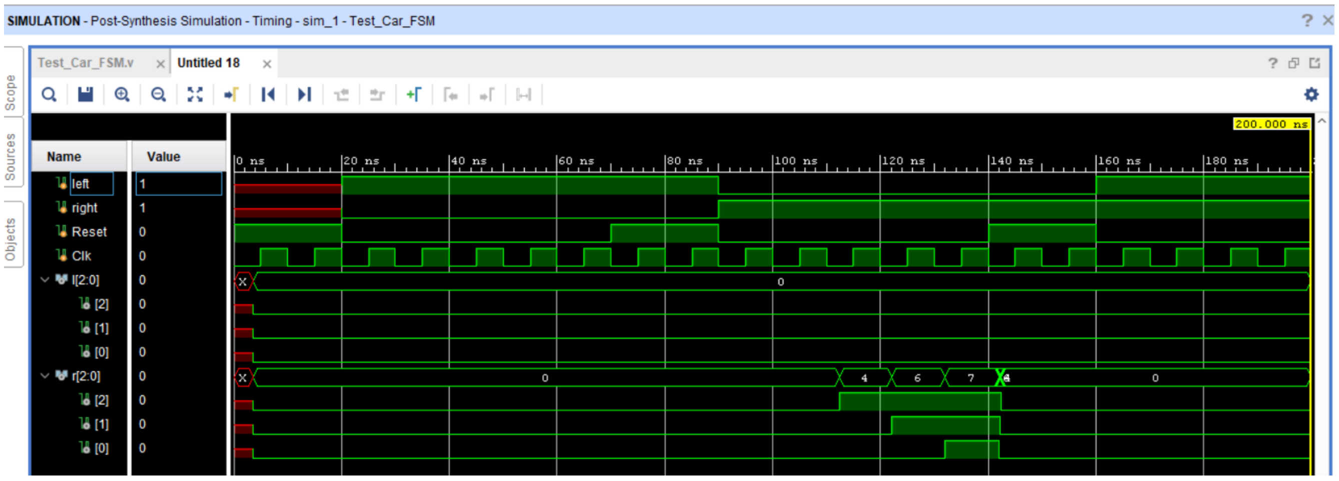
   Question: Paste the image showing the schematic after synthesis.

   Answer:



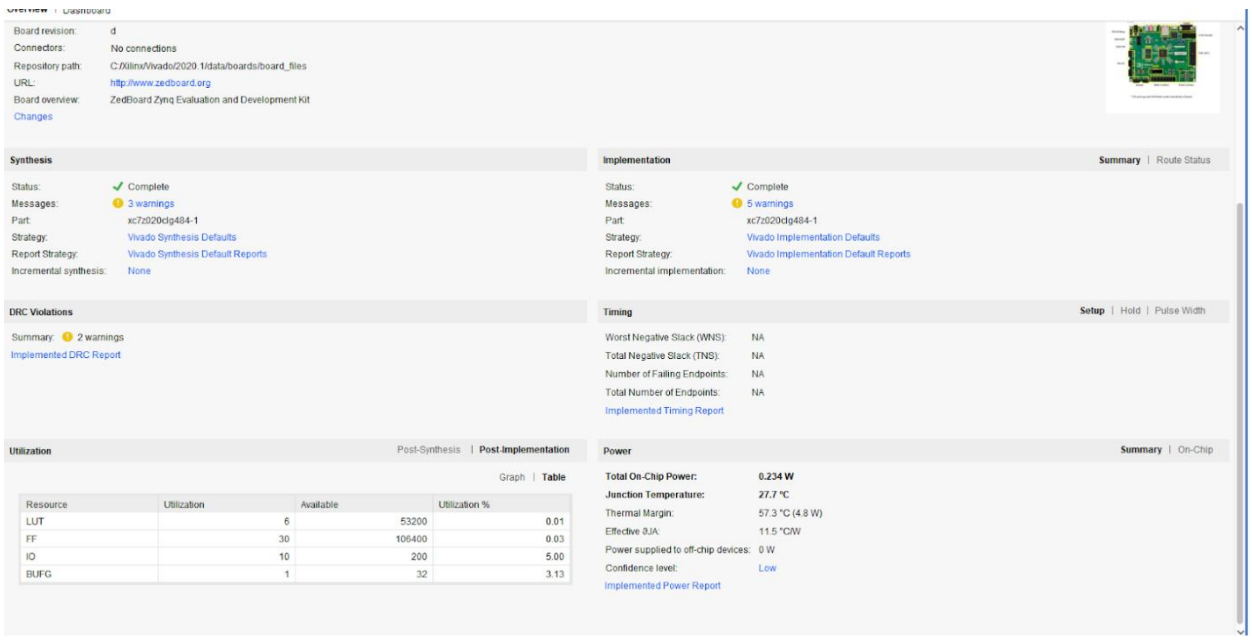
Question: Paste the image showing the Post-Synthesis Simulation for FSM.

Answer:



Question: Check the summary report and report hardware utilization for the FSM implementation.

Answer:



Overview | Dashboard

Board revision:	d
Connectors:	No connections
Repository path:	C:/Xilinx/Vivado/2020.1/data/boards/board_files
URL:	http://www.zedboard.org
Board overview:	ZedBoard Zynq Evaluation and Development Kit

Changes

Synthesis

Status:	✓ Complete
Messages:	3 warnings
Part:	xc7z020clg484-1
Strategy:	Vivado Synthesis Defaults
Report Strategy:	Vivado Synthesis Default Reports
Incremental synthesis:	None

Implementation — Summary | Route Status

Status:	✓ Complete
Messages:	5 warnings
Part:	xc7z020clg484-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None

DRC Violations

Summary: 2 warnings

Implemented DRC Report

Timing — Setup | Hold | Pulse Width

Worst Negative Slack (WNS):	NA
Total Negative Slack (TNS):	NA
Number of Failing Endpoints:	NA
Total Number of Endpoints:	NA

Implemented Timing Report

Utilization — Post-Synthesis | Post-Implementation

Graph | Table

Resource	Utilization	Available	Utilization %
LUT	6	53200	0.01
FF	30	106400	0.03
IO	10	200	5.00
BUFG	1	32	3.13

Power — Summary | On-Chip

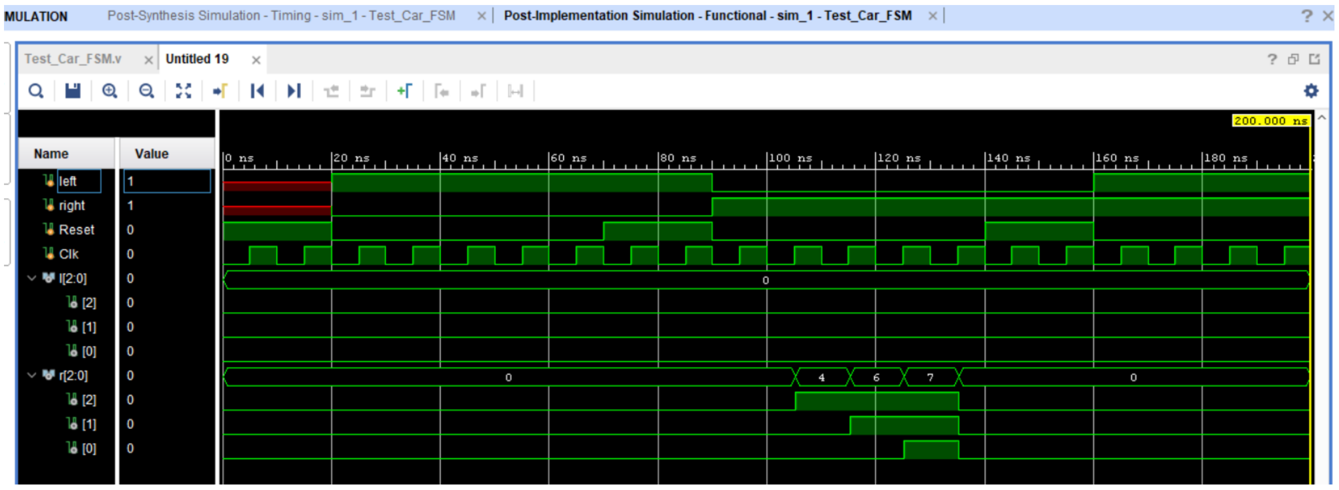
Total On-Chip Power:	**0.234 W**
Junction Temperature:	**27.7 °C**
Thermal Margin:	57.3 °C (4.8 W)
Effective ϑJA:	11.5 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Implemented Power Report

7. Implement the design (**Run Implementation**).

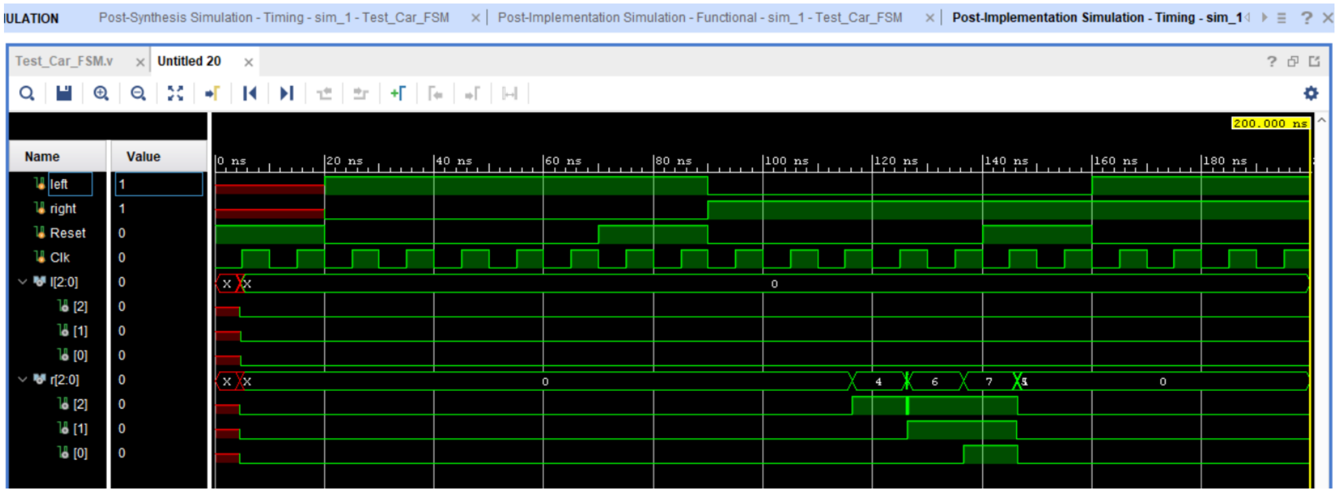
Question: Paste the image showing the Post-Implementation Simulation for FSM.

Answer:


Post-Synthesis Simulation - Timing - sim_1 - Test_Car_FSM
Post-Implementation Simulation - Functional - sim_1 - Test_Car_FSM
Test_Car_FSM.v
Untitled 19
Name
Value
left
right
Reset
Clk
l[2:0]
r[2:0]
200.000 ns

Question: Paste the image showing the Post-Implementation timing Simulation for FSM.

Answer:


Post-Synthesis Simulation - Timing - sim_1 - Test_Car_FSM
Post-Implementation Simulation - Functional - sim_1 - Test_Car_FSM
Post-Implementation Simulation - Timing - sim_1
Test_Car_FSM.v
Untitled 20
Name
Value
left
right
Reset
Clk
l[2:0]
r[2:0]
200.000 ns

Question: Justify the difference between Post-Implementation and Post-Implementation timing Simulation waveforms.

Answer: Timing simulation is to verify whether the circuit will work at the required speed. Functional simulation is to verify the design of our circuit.

8. **Generate Bitstream** and port your design on to FPGA (**Open Hardware Manager New Target… Program Device**)

9. **Check the output on FPGA.**

10. Show the output to the instructor.

11. Submit following files as a Zipped folder with file name as <Student1_ID_No>_<Student2_ID_No>.zip through CMS before due date.

1) Completed Document
2) Car_FSM.v (with proper comments)
3) Test_Car_FSM.v (with proper comments)
4) Car_FSM.xdc.